

# КЛІЄНТ-СЕРВЕРНА ПЛАТФОРМА НА БАЗІ ARM ПРОЦЕСОРУ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ Ktor

Вінницький національний технічний університет

## Анотація

*Розробка клієнт-серверної платформи на базі ARM процесору за допомогою якої можна обмінюватись даними (команди, повідомлення, інші дані) між пристроями через сервер, використовуючи високо ефективні міні-комп'ютери на ARM процесорі. Вся платформа реалізована на мові програмування Kotlin(JVM) що дозволяє даному рішенню бути мультиплатформовим.*

**Ключові слова:** мультиплатформовий, kotlin, ktor, платформа

## Abstract

*Development of a client-server platform based on the ARM processor with which you can exchange data (commands, messages, other data) between devices through the server, using highly efficient mini-computers on the ARM processor. The whole platform is implemented in the Kotlin programming language (JVM), which allows this solution to be multiplatform.*

**Keywords:** multiplatform, kotlin, ktor, platform.

## Вступ

На сьогоднішній день більшість компаній і організацій мають свої клієнт-серверні платформи наподоби: стримінгових сервісів Netflix, Spotify, YouTube, сервісів по керуванню пристроями Mi Home, Blynk і т.д, всі вони мають схожий принцип роботи. Клієнт-серверна платформа складається з клієнтів у вигляді: мобільних пристроїв, ПК, телевізорів, “розумних” пристроїв і серверної сторони, яка може бути як хмарним веб сервісом які надають постачальники послуг як AmazonAWS, так і конкретними фізичними серверами в дата-центрі або в якомусь приміщенні. Кожен клієнт має певну операційну систему як Android, IOS, Windows, і певний додаток для роботи і взаємодії з сервісом. В більшості випадків для кожної операційної системи створюється додаток окремо, відповідно використовуються різні мови програмування, бібліотеки і фреймворки, через що ускладнюється підтримка цих додатків. Також програмне забезпечення на серверній стороні повинно підтримувати коректний формат зв'язку (передачу даних) між сервером і клієнтом, що накладає додаткові проблеми при налагодженні і впровадженні нових функцій в платформу.

## Результат дослідження

Kotlin - статично типізований, об'єктно-орієнтована, мультиплатформенна мова програмування, що працює поверх Java Virtual Machine і розробляється компанією JetBrains. Автори ставили за мету створити мову більш лаконічну і типобезпечну, ніж Java, і більш просту, ніж Scala. Наслідком спрощення в порівнянні зі Scala стали також більш швидка компіляція і найкраща підтримка мови в IDE. Мова повністю сумісна з Java, що дозволяє java-розробникам поступово перейти до його використання; зокрема, в Android мова вбудовується за допомогою Gradle, що дозволяє для існуючого android-додатка впроваджувати нові функції на Kotlin без переписування програми цілком.

Ktor - це фреймворк, який дозволяє легко створювати підключені програми - веб-програми, служби HTTP, мобільні та браузерні програми, сервери і т.д. Підтримує всі можливості мови програмування Kotlin. В даний час підтримуються сценарії клієнта та сервера JVM, а також клієнти JavaScript, iOS, Android, Windows та Linux.

Корутини(Kotlin) - це полегшені потоки. Полегшений потік означає, що він не прив'язаний до нативному потоку, тому він не вимагає перемикання контексту на процесор, тому він швидше. Корутини

можна представити у вигляді полегшеного потоку. Подібно потокам, корутини можуть працювати паралельно, чекати один одного і спілкуватися. Найбільша відмінність полягає в тому, що корутини дуже дешеві, майже безкоштовні: ми можемо створювати їх тисячами і платити дуже мало з точки зору продуктивності. Потоки ж обходяться дорого. Тисяча потоків може стати серйозною проблемою навіть для сучасної машини.

На стороні сервера створюється проект асинхронного сервера використовуючи Ktor, дані передаються через http/https запити, кожен запит обробляється асинхронно за допомогою корутин, логування користувача відбувається за допомогою влаштованого модуля Ktor аутентифікації і авторизації, дані користувача зберігаються в базі даних mongoDB, а також тимчасові дані які потребують велику швидкість запису і читання, працюють у віртуальній базі даних в оперативній пам'яті, після чого записуються в mongoDB. В ролі сервера використовується Raspberry Pi 4, на базі 4 ядерного ARM процесора з операційною системою Linux Ubuntu 20.04, оскільки мова програмування виконується на віртуальній машині джава (JVM) сервер без проблем запускається. На стороні клієнта також використовується Ktor де тим же способом відбуваються запити на сервер, а також влаштований модуль аутентифікації і авторизації. Графічна частина UI клієнта написана на JetPack Compose Multiplatform.

Результат - отримано високопродуктивну платформу яка може одночасно підтримувати тисячі клієнтів, при малому енергоспоживанні кластера серверів, а також підтримка на стороні клієнта операційних систем: Windows, Linux, MacOS, Android, IOS.

**Важливо:** сервер буде працювати тільки з версії JVM11.

**Переваги клієнт-серверної платформи на базі ARM процесору з використанням технології Ktor:**

1. Зручна і швидка розробка додатків як на стороні клієнта, так і на серверній стороні;
2. Можна використовуючи велику кількість спільного коду програм;
3. Майже весь код платформи написаний на мові програмування Kotlin, що пришвидшує розробку, і зменшує кількість проблем;
4. Підтримка більшості популярних операційних систем;
5. Швидкість обробки запитів сервером менше 1 секунди (включно з запитами які потребують взаємодію з базою даних);
6. Мале енергоспоживання серверів і простота обслуговування;
7. Мале навантаження на сервер при великій кількості запитів.

#### **Висновок**

У даній роботі було показано реалізацію вискоєфективної клієнт-серверної платформи, з великою швидкістю розробки.

#### **СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Документація Kotlin [Електронний ресурс] – Режим доступу до ресурсу: <https://kotlinlang.org/> (дата звернення: 8.03.2021).
2. Документація Ktor [Електронний ресурс] – Режим доступу до ресурсу: <https://ktor.io/docs/> (дата звернення: 8.03.2021).
3. Фреймворк Compose for Desktop [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/jetbrains/compose-jb> (дата звернення: 8.03.2021).

Станіславенко Максим Михайлович, студент групи ІКІ-19МС, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: maxappcomp@gmail.com

Науковий керівник: Богомолів Сергій Віталійович, к.т.н., доцент каф. ОТ, Вінницький національний технічний університет, м. Вінниця, e-mail: bogomolovsergiy@vntu.edu.ua

Stanislavenko Maksym, student of the group ICE-19MS, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: maxappcomp@gmail.com

Supervisor: Bohomolov Serhii, Ph.D., Ph.D., Associate Professor kaf. OT, Vinnytsia National Technical University, Vinnytsia, e-mail: bogomolovsergiy@vntu.edu.ua